

# 1 Aufgabe UE-1

## 1.1 Aufgabe UE-1.1

Schreiben Sie eine Matlab-Funktion  $[w] = \text{perco}(X, t, \text{maxEpoches})$ , welche den Gewichtsvektor  $w$  eines Perceptrons für das Trainingsset  $(X, t)$  mittels des im Skriptum beschriebenen online-Verfahrens bestimmt. Das Trainingsset soll dabei wie im Skriptum beschrieben aufgebaut sein ( $i$ -te Spalte enthält  $i$ -tes Muster  $X(:,i)$  bzw. Target (Klassenlabel)  $t(i)$ ,  $+1/-1$  - Kodierung der Targets). In  $\text{maxEpoches}$  soll zusätzlich die maximale Anzahl der Epochen übergeben werden können.

Wenden Sie das Verfahren auf folgende Probleme an: OR, AND, XOR. Vergessen Sie nicht, Ihre Trainingsvektoren in homogenen Koordinaten zu übergeben.

## 1.2 Aufgabe UE-1.2

Trainieren Sie Ihr Perceptron auf den Inputvektoren in der Datei *perceptron-data* mit den Targets (Klassen-Labels) *perceptrontarget1* und *perceptrontarget2*. Plotten Sie außerdem die Klassenaufteilung der Trainingspunkte für beide Mengen von Klassen-Labels.

Hinweise:

- Die Dateien können mittels der Matlab-Funktion *load* geladen werden, z.B. *inputs = load('perceptrondata')*. Beachten Sie, dass die Trainingsmuster / Targets zeilenweise gespeichert sind.
- Die Targets sind 0/1-codiert. Dies stimmt mit der in der NN-Toolbox verwendeten Konvention überein; um jedoch ihr in UE-1.1 entwickeltes Verfahren verwenden zu können, müssen Sie die Targets zunächst in -1/+1-Codierung 'konvertieren'.
- Um das Trainingsset in positive und negative Punkte zu trennen, ist es hilfreich, mit logischen Indizes zu arbeiten. Enthält z.B. der Zeilenvektor  $t$  in  $t(i)$  die Klassenzugehörigkeit des  $i$ -ten Musters  $X(:,i)$ , so erhält man mittels:  $\text{logind} = (t == 1)$  einen logischen Index, der genau jene Punkte aus  $X$  auswählt, deren Target 1 ist, z.B.:  $\text{patterns1} = X(:, \text{logind})$ .

## 2 Aufgabe UE-2

Generieren Sie je 100 Stichproben (*samples*) der Größe 5/30/100/500 für die Normalverteilung  $N(\mu, \sigma^2) = N(4, 3)$ . Hierzu steht unter MATLAB die Funktion *normrnd* zu Verfügung. Beachten Sie jedoch, dass viele Funktionen der Statistik-Toolbox die Standardabweichung  $\sigma$  und nicht die Varianz  $\sigma^2$  als Argument erwarten.

Wie verteilen sich für jede der 4 Stichprobengrößen die 100 geschätzten Mittelwerte? Stellen Sie deren Verteilung in Histogrammform dar (Funktion *hist*), berechnen Sie das empirische Mittel sowie die empirische Varianz und vergleichen Sie Ihre Ergebnisse mit den theoretischen Resultaten bzgl. der Verteilung des Mittelwerts. Mittelwert und Varianz einer Zahlenfolge können unter MATLAB mittels *mean* bzw. *var* berechnet werden.